31.9 – Considerando que o sistema é orientado à pesquisa e desenvolvimento e tem por objetivo produzir um produto no próximo ano, será utilizado um TSP (Team Software Process), sob uma perspectiva evolucionária e utilizando os scritps do desenvolvimento ágil. A organização utilizará o paradigma síncrono, dada a especificidade do problema, no qual ocorre uma natural compartimentalização das atividades de desenvolvimento conforme as habilidades de seus membros.

32.6 – Quantidade de linhas de código não deve ser considerada, isoladamente, como critério para avaliação de produtividade. Por exemplo, tempo de execução de um algoritmo não pode ser inferido, diretamente, pelo número de linhas de código. Talvez um código longo, vinculado a uma abordagem ineficiente na resolução do problema, visto sob o prisma da quantidade de linhas de código, seja muito produtivo. No entanto, no que diz respeito a qualidade do software desenvolvimento, pesará negativamente.

33.2 – Complexidade do problema a ser solucionado;

Complexidade dos algoritmos a serem utilizados;

Complexidade dos conhecimentos necessários para entender o problema e apresentar uma solução viável.

34.5 – Caso a pessoa a ser introduzida tenha vasto conhecimento nas metodologias utilizadas no projeto; tenha vasto conhecimento sobre a área na qual projeto busca solução; ou seja, caso a bagagem de experiência que essa pessoa traga seja ampla o suficiente para que o seu tempo de adaptação ao projeto seja muito curto, ou inexistente, a introdução dela ao time produzirá ganhos de desempenho.

35.5 –

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Componentes  Categorias | Performance | Suporte | Custo | Cronograma |
| Catastrófico | O sistema toma mais tempo que a concorrência para apresentar uma solução viável |  | O custo para execução excede 70% do retorno esperado com o projeto. | O tempo necessário para aquisição do conhecimento necessário excede o tempo médio das mudanças tecnológicas da área. |
| Crítico |  |  |  |  |
| Marginal |  |  |  |  |
| Negligenciável |  |  |  |  |
|  |  |  |  |  |

36.6 – No processo de manutenibilidade e reengenharia, é difícil, dada a complexidade do projeto envolvido, conseguir reconstruir, em nível elevado de abstração, quais os propósitos de uma determinada parte do código e sua iteração com as demais partes do sistema como um todo. Ou seja, a imagem macro do sistema, como concebida na mente de seus criadores, por mais que tenha sido traduzida em código, no sentido inverso (do código para a ideia), talvez até mesmo se executado por seu criador, perderá alguns detalhes da imagem quando comparada à original.